Introduction to Machine Learning

Feature Generation

Andres Mendez-Vazquez

February 25, 2019

Outline

Outline

What do we want?

What

Given a set of measurements, the goal is to discover compact and informative representations of the obtained data.

Our Approach

We want to "squeeze" in a relatively small number of features, leading to a reduction of the necessary feature space dimension.

Properties

Thus removing information redundancies - Usually produced and the measurement.

What do we want?

What

Given a set of measurements, the goal is to discover compact and informative representations of the obtained data.

Our Approach

We want to “squeeze” in a relatively small number of features, leading to a reduction of the necessary feature space dimension.

Properties

Thus removing information redundancies - Usually produced and the measurement.

What do we want?

What

Given a set of measurements, the goal is to discover compact and informative representations of the obtained data.

Our Approach

We want to "squeeze" in a relatively small number of features, leading to a reduction of the necessary feature space dimension.

Properties

Thus removing information redundancies - Usually produced and the measurement.

What Methods we will see?

Fisher Linear Discriminant

1. Squeezing to the maximum.
2. From Many to One Dimension

Principal Component Analysis

1. Not so much squeezing
2. You are willing to lose some information

What Methods we will see?

Fisher Linear Discriminant

1. Squeezing to the maximum.
2. From Many to One Dimension

Principal Component Analysis

1. Not so much squeezing
2. You are willing to lose some information

However, Please review

Singular Value Decomposition

1. Decompose a $m \times n$ data matrix A into $A = USV^T$, U and V orthonormal matrices and S contains the eigenvalues.
2. You can read more of it on “Singular Value Decomposition Tutorial” at the paper section.

Outline

Rotation

Projecting

Projecting well-separated samples onto an arbitrary line usually produces a confused mixture of samples from all of the classes and thus produces poor recognition performance.

Something Notable

However, moving and rotating the line around might result in an orientation for which the projected samples are well separated.

Fisher linear discriminant (FLD)

It is a discriminant analysis seeking directions that are efficient for discriminating binary classification problem.

Rotation

Projecting

Projecting well-separated samples onto an arbitrary line usually produces a confused mixture of samples from all of the classes and thus produces poor recognition performance.

Something Notable

However, moving and rotating the line around might result in an orientation for which the projected samples are well separated.

Fisher linear discriminant (FLD)

It is a discriminant analysis seeking directions that are efficient for discriminating binary classification problem.

Rotation

Projecting

Projecting well-separated samples onto an arbitrary line usually produces a confused mixture of samples from all of the classes and thus produces poor recognition performance.

Something Notable

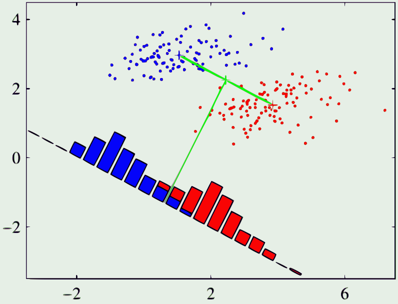
However, moving and rotating the line around might result in an orientation for which the projected samples are well separated.

Fisher linear discriminant (FLD)

It is a discriminant analysis seeking directions that are efficient for discriminating binary classification problem.

Example

Example - From Left to Right the Improvement



This is actually comming from...

Classifier as

A machine for dimensionality reduction.

Initial Setup

We have:

- N d-dimensional samples $x_1, x_2, ..., x_N$
- N_i is the number of samples in class C_i for i=1,2.

Then, we ask for the projection of each x_i into the line by means of

$$y_i = w^T x_i \tag{1}$$

This is actually comming from...

Classifier as

A machine for dimensionality reduction.

Initial Setup

We have:

- N d-dimensional samples $x_1, x_2, ..., x_N$
- N_i is the number of samples in class C_i for i=1,2.

Then, we ask for the projection of each x_i into the line by means of

$$y_i = w^T x_i \tag{1}$$

This is actually comming from...

Classifier as

A machine for dimensionality reduction.

Initial Setup

We have:

- N d-dimensional samples $x_1, x_2, ..., x_N$
- N_i is the number of samples in class C_i for i=1,2.

Then, we ask for the projection of each x_i into the line by means of

$$y_i = \boldsymbol{w}^T \boldsymbol{x}_i \tag{1}$$

Outline

Use the mean of each Class

Then

Select $\boldsymbol{w}$ such that class separation is maximized

We then define the mean sample for ecah class

1. $C_1 \Rightarrow \boldsymbol{m}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} \boldsymbol{x}_i$
2. $C_2 \Rightarrow \boldsymbol{m}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} \boldsymbol{x}_i$

Ok!!! This is giving us a measure of distance

Thus, we want to maximize the distance the projected means:

$$m_1 - m_2 = \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{2}$$

where $m_k = \boldsymbol{w}^T \boldsymbol{m}_k$ for $k = 1, 2$.

Use the mean of each Class

Then

Select $\boldsymbol{w}$ such that class separation is maximized

We then define the mean sample for ecah class

1. $C_1 \Rightarrow \boldsymbol{m}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} \boldsymbol{x}_i$
2. $C_2 \Rightarrow \boldsymbol{m}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} \boldsymbol{x}_i$

Ok!!! This is giving us a measure of distance

Thus, we want to maximize the distance the projected means:

$$m_1 - m_2 = \boldsymbol{w}^T\left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right) \tag{2}$$

where $m_k = \boldsymbol{w}^T \boldsymbol{m}_k$ for $k = 1, 2$.

Use the mean of each Class

Then

Select $\boldsymbol{w}$ such that class separation is maximized

We then define the mean sample for ecah class

1. $C_1 \Rightarrow \boldsymbol{m}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1}\boldsymbol{x}_i$
2. $C_2 \Rightarrow \boldsymbol{m}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2}\boldsymbol{x}_i$

Ok!!! This is giving us a measure of distance

Thus, we want to maximize the distance the projected means:

$$m_1 - m_2 = \boldsymbol{w}^T\left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right) \tag{2}$$

where $m_k = \boldsymbol{w}^T\boldsymbol{m}_k$ for $k = 1, 2$.

However

We could simply seek

$$
\begin{aligned}
\max \boldsymbol{w}^T \left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right) \\
s.t. \sum_{i=1}^{d} w_i = 1
\end{aligned}
$$

After all

We do not care about the magnitude of w.

However

We could simply seek

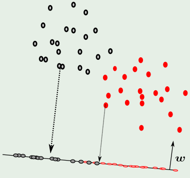
$$\max \boldsymbol{w}^T \left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)$$
$$s.t. \sum_{i=1}^{d} w_i = 1$$

After all

We do not care about the magnitude of $\boldsymbol{w}$.

Example

Here, we have the problem



Outline

Fixing the Problem

To obtain good separation of the projected data

The difference between the means should be large relative to some measure of the standard deviations for each class.

We define a SCATTER measure (Based in the Sample Variance)

$$s_k^2 = \sum_{x_i \in C_k} \left(w^T x_i - m_k \right)^2 = \sum_{y_i = w^T x_i \in C_k} \left(y_i - m_k \right)^2 \tag{3}$$

We define then within-class variance for the whole data

$$s_1^2 + s_2^2 \tag{4}$$

Fixing the Problem

To obtain good separation of the projected data

The difference between the means should be large relative to some measure of the standard deviations for each class.

We define a SCATTER measure (Based in the Sample Variance)

$$s_k^2 = \sum_{\boldsymbol{x}_i \in C_k} \left(\boldsymbol{w}^T \boldsymbol{x}_i - m_k \right)^2 = \sum_{y_i = \boldsymbol{w}^T \boldsymbol{x}_i \in C_k} \left(y_i - m_k \right)^2 \tag{3}$$

We define then within-class variance for the whole data

$$s_1^2 + s_2^2 \tag{4}$$

Fixing the Problem

To obtain good separation of the projected data

The difference between the means should be large relative to some measure of the standard deviations for each class.

We define a SCATTER measure (Based in the Sample Variance)

$$s_k^2 = \sum_{\boldsymbol{x}_i \in C_k} \left(\boldsymbol{w}^T \boldsymbol{x}_i - m_k \right)^2 = \sum_{y_i = \boldsymbol{w}^T \boldsymbol{x}_i \in C_k} (y_i - m_k)^2 \tag{3}$$

We define then within-class variance for the whole data

$$s_1^2 + s_2^2 \tag{4}$$

Outline

Finally, a Cost Function

The between-class variance

$$(m_1 - m_2)^2 \tag{5}$$

The Fisher criterion

$$\frac{\text{between-class variance}}{\text{within-class variance}} \tag{6}$$

Finally

$$J(w) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \tag{7}$$

Finally, a Cost Function

The between-class variance

$$(m_1 - m_2)^2 \tag{5}$$

The Fisher criterion

$$\frac{\text{between-class variance}}{\text{within-class variance}} \tag{6}$$

Finally

$$J(w) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \tag{7}$$

Finally, a Cost Function

The between-class variance

$$(m_1 - m_2)^2 \tag{5}$$

The Fisher criterion

$$\frac{\text{between-class variance}}{\text{within-class variance}} \tag{6}$$

Finally

$$J(\boldsymbol{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \tag{7}$$

From it, we can obtain

An approximation to the $\boldsymbol{w}$

$$\boldsymbol{w} \propto \boldsymbol{S}_w^{-1} (\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{8}$$

From it, we can obtain

An approximation to the $\boldsymbol{w}$

$$\boldsymbol{w} \propto \boldsymbol{S}_w^{-1} \left(\boldsymbol{m}_1 - \boldsymbol{m}_2 \right) \tag{8}$$

Once the data is transformed into y_i

- Use a threshold $y_0 \Rightarrow x \in C_1$ iff $y(x) \geq y_0$ or $x \in C_2$ iff $y(x) < y_0$
- Or ML with a Gussian can be used to classify the new transformed data using a Naive Bayes (Central Limit Theorem and $y = w^T x$ sum of random variables).

From it, we can obtain

An approximation to the $\boldsymbol{w}$

$$\boldsymbol{w} \propto \boldsymbol{S}_w^{-1} \left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right) \tag{8}$$

Once the data is transformed into y_i

- Use a threshold $y_0 \Rightarrow x \in C_1$ iff $y(x) \geq y_0$ or $x \in C_2$ iff $y(x) < y_0$
- Or ML with a Gussian can be used to classify the new transformed data using a Naive Bayes (Central Limit Theorem and $y = \boldsymbol{w}^T \boldsymbol{x}$ sum of random variables).

Please

Your Reading Material, it is about the Multiclass

4.1.6 Fisher's discriminant for multiple classes AT “Pattern Recognition” by Bishop

Outline

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Goal

Project data onto space with dimensionality $m < d$ (We assume m is given)

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Goal

Project data onto space with dimensionality $m < d$ (We assume m is given)

Dimensional Variance

Remember the Sample Variance Sample

$$VAR(X) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})}{N-1} \tag{9}$$

You can do the same in the case of two variables X and Y

$$COV(X, Y) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{N-1} \tag{10}$$

Dimensional Variance

Remember the Sample Variance Sample

$$VAR(X) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})}{N-1} \tag{9}$$

You can do the same in the case of two variables X and Y

$$COV(X,Y) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{N-1} \tag{10}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{11}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{12}$$

Build new data

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{13}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{11}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{12}$$

Build new data

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{13}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{11}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \tag{12}$$

Build new data

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{13}$$

Build the Sample Mean

The Covariance Matrix

$$S = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{14}$$

Properties

1. The ijth value of S is equivalent to σ_{ij}^2.
2. The iith value of S is equivalent to σ_{ii}^2.
3. What else? Look at a plane Center and Rotating!!!

Build the Sample Mean

The Covariance Matrix

$$S = \frac{1}{N-1} \sum_{i=1}^{N} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T \tag{14}$$

Properties

1. The ijth value of S is equivalent to σ_{ij}^2.
2. The iith value of S is equivalent to σ_{ii}^2.
3. What else? Look at a plane Center and Rotating!!!

Outline

Using S to Project Data

As in Fisher

We want to project the data to a line...

For this we use a u_1

with $u_1^T u_1 = 1$

Question

What is the Sample Variance of the Projected Data

Using S to Project Data

As in Fisher

We want to project the data to a line...

For this we use a $\boldsymbol{u}_1$

with $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$

Question

What is the Sample Variance of the Projected Data

Using S to Project Data

As in Fisher

We want to project the data to a line...

For this we use a $\boldsymbol{u}_1$

with $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$

Question

What is the Sample Variance of the Projected Data

Outline

Thus we have

Variance of the projected data

$$\frac{1}{N-1}\sum_{i=1}^{N}\left[\boldsymbol{u}_1\boldsymbol{x}_i-\boldsymbol{u}_1\overline{\boldsymbol{x}}\right]=\boldsymbol{u}_1^T S\boldsymbol{u}_1 \tag{15}$$

Use Lagrange Multipliers to Maximize

$$\boldsymbol{u}_1^T S\boldsymbol{u}_1+\lambda_1\left(1-\boldsymbol{u}_1^T\boldsymbol{u}_1\right) \tag{16}$$

Thus we have

Variance of the projected data

$$\frac{1}{N-1}\sum_{i=1}^{N}\left[\boldsymbol{u}_1\boldsymbol{x}_i - \boldsymbol{u}_1\overline{\boldsymbol{x}}\right] = \boldsymbol{u}_1^T S \boldsymbol{u}_1 \tag{15}$$

Use Lagrange Multipliers to Maximize

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 + \lambda_1\left(1 - \boldsymbol{u}_1^T\boldsymbol{u}_1\right) \tag{16}$$

Derive by $\boldsymbol{u}_1$

We get

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{17}$$

Then

$\boldsymbol{u}_1$ is an eigenvector of S.

If we left-multiply by $\boldsymbol{u}_1$

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{18}$$

Derive by $\boldsymbol{u}_1$

We get

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{17}$$

Then

$\boldsymbol{u}_1$ is an eigenvector of S.

If we left-multiply by $\boldsymbol{u}_1$

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{18}$$

Derive by $\boldsymbol{u}_1$

We get

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{17}$$

Then

$\boldsymbol{u}_1$ is an eigenvector of S.

If we left-multiply by $\boldsymbol{u}_1$

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{18}$$

What about the second eigenvector $\boldsymbol{u}_2$

We have the following optimization problem

$$\begin{aligned} \max \quad & \boldsymbol{u}_2^T S \boldsymbol{u}_2 \\ \text{s.t.} \quad & \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1 \\ & \boldsymbol{u}_2^T \boldsymbol{u}_1 = 0 \end{aligned}$$

Lagrangian

$$L\left(\boldsymbol{u}_2, \lambda_1, \lambda_2\right) = \boldsymbol{u}_2^T S \boldsymbol{u}_2 - \lambda_2 \left(\boldsymbol{u}_2^T \boldsymbol{u}_2 - 1\right) - \lambda_1 \left(\boldsymbol{u}_2^T \boldsymbol{u}_1 - 0\right)$$

What about the second eigenvector $\boldsymbol{u}_2$

We have the following optimization problem

$$
\begin{aligned}
\max \ & \boldsymbol{u}_2^T S \boldsymbol{u}_2 \\
\text{s.t.} \ & \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1 \\
& \boldsymbol{u}_2^T \boldsymbol{u}_1 = 0
\end{aligned}
$$

Lagrangian

$$
L\left(\boldsymbol{u}_2, \lambda_1, \lambda_2\right) = \boldsymbol{u}_2^T S \boldsymbol{u}_2 - \lambda_2 \left(\boldsymbol{u}_2^T \boldsymbol{u}_2 - 1\right) - \lambda_1 \left(\boldsymbol{u}_2^T \boldsymbol{u}_1 - 0\right)
$$

With Solution

We have

$$\boldsymbol{u}_2^T S \boldsymbol{u}_2 = \lambda_2$$

Implying

- u_2 is the eigenvector of S with second largest eigenvalue λ_2.

With Solution

We have

$$\boldsymbol{u}_2^T S \boldsymbol{u}_2 = \lambda_2$$

Implying

- $\boldsymbol{u}_2$ is the eigenvector of S with second largest eigenvalue λ_2.

Thus

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{19}$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Computational Cost

1. Full eigenvector decomposition $O\left(d^3\right)$
2. Power Method $O\left(Md^2\right)$ "Golub and Van Loan, 1996)"
3. Use the Expectation Maximization Algorithm

Thus

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{19}$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Computational Cost

1. Full eigenvector decomposition $O\left(d^3\right)$
2. Power Method $O\left(Md^2\right)$ "Golub and Van Loan, 1996)"
3. Use the Expectation Maximization Algorithm

Thus

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{19}$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Computational Cost

1. Full eigenvector decomposition $O\left(d^3\right)$
2. Power Method $O\left(Md^2\right)$ "Golub and Van Loan, 1996)"
3. Use the Expectation Maximization Algorithm

Outline

We have the following steps

Determine covariance matrix

$$S = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{20}$$

Generate the decomposition

$$S = U\Sigma U^T$$

With

- Eigenvalues in Σ and eigenvectors in the columns of U.

We have the following steps

Determine covariance matrix

$$S = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{20}$$

Generate the decomposition

$$S = U\Sigma U^T$$

With

- Eigenvalues in Σ and eigenvectors in the columns of U.

We have the following steps

Determine covariance matrix

$$S = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{20}$$

Generate the decomposition

$$S = U\Sigma U^T$$

With

- Eigenvalues in Σ and eigenvectors in the columns of U.

Then

Project samples $\boldsymbol{x}_i$ into subspaces dim$=k$

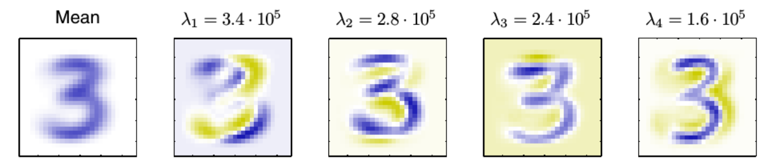
$$z_i = U_K^T \boldsymbol{x}_i$$

- With U_k is a matrix with k columns

Outline

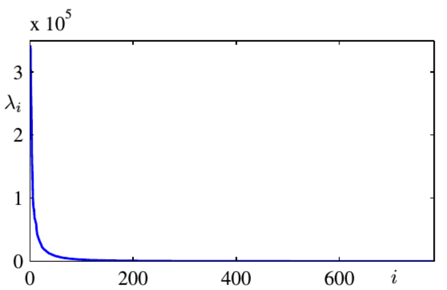
Example

From Bishop



Example

From Bishop



Example

From Bishop